2017 SCHOOL ON BRAIN CONNECTOMICS

Hands-on Session 12.10.2017:

Brain Connectivity and Graphs

Alessandra Griffa, PhD

alessandra.griffa@gmail.com

Dutch Connectome Lab – University Medical Center Utrecht

In this hands-on session you will use MATLAB to compute fundamental brain networks features and investigate some aspects of brain topology. The hands-on is divided in four parts which deals with the following topics:

1. Network adjacency matrices: visualization and interpretation
2. Measures of integration and segregation in small-world networks
3. Brain hubs and network visualization
4. Group-comparison for clinical studies and statistical issues

This booklet will guide you through a series of exercises. No advanced MATLAB knowledge is required, and MATLAB tips-and-tricks will be provided to facilitate the coding part. In the exercises you will also find some questions to help the discussion in the group.

In the exercises you will use some handy MATLAB toolboxes dedicated to brain connectivity analysis. You can get extra information on these toolboxes by visiting the following resources:

Brain Connectivity Toolbox (BCT)

https://sites.google.com/site/bctnet/

This is a collection of MATLAB functions that compute graph theoretical measures and implement a series of algorithms such as shortest path identification or network community detection. The graph measures and algorithms implemented in this toolbox are described in the paper '*Complex network measures of brain connectivity: Uses and interpretations*' (Rubinov and Sporns, 2010). During the lab, it will be useful to have a look at the table at the end of the paper, which reports the mathematical expressions of different graph measures. You can find the pdf of the paper in the folder '*Lab_Brain_Connectivity_And_Graphs/Resources*'.

BrainNet Viewer

https://www.nitrc.org/projects/bnv

This is a MATLAB toolbox (with user interface) for 3D visualization of brain networks (for the manual: '*Lab_Brain_Connectivity_And_Graphs/Utilities/ ... BrainNetViewer_20170403/BrainNet_Manual.pdf*').

Getting started

To start this hands-on session, copy the folder '*Lab_Brain_Connectivity_And_Graphs*' to your Desktop.
Start MATLAB and change the MATLAB working directory to:
'*...\Desktop\Lab_Brain_Connectivity_And_Graphs*'.
To change the MATLAB working directory you can type the following command at the MATLAB command line (insert your path):

```
>> cd('...\Desktop\Lab_Brain_Connectivity_And_Graphs')
```

Add the folder '*Utilities*' and its subfolders to your MATLAB path. '*Utilities*' contains the BrainNet Viewer, the BCT toolbox and some additional functions you will use in this lab.

```
>> addpath(genpath('Utilities'))
```

As a general remark, if you have any doubt concerning the usage, syntax or meaning of a function you can type

```
>> doc function_name
```

to visualize the relative documentation. For example, check out what the function '*genpath*' does by typing:

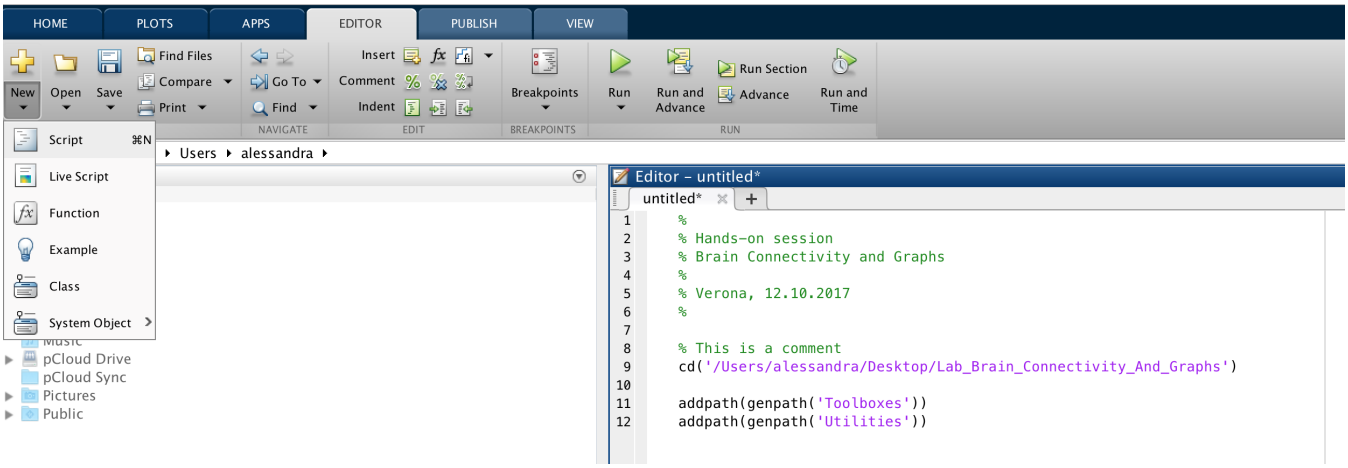
```
>> doc genpath
```

MATLAB scripting

In programming languages, *scripts* are text files containing sequences of instructions. A script stores commands exactly as you would type them at the command line.
In this hands-on session it will be convenient to store your commands in a MATLAB script. In this way you will be able to re-use the code at your convenience.

To initiate a new MATLAB script click the '*New Script*' button in the *Editor* tab. Write the commands you have just learnt ('*cd*', '*addpath*') in the script *Editor*. Note that you can also insert comments in the script using the symbol '*%*'. Save your script clicking the '*Save*' button and using the MATLAB scripts' extension '*.m*'.

You can now execute your script by clicking the button '*Run*' in the *Editor* tab. This will sequentially execute all the lines of code in the script, one after the other.
In the hands-on session it will be more convenient to execute only one or few instructions at a time. To execute only one or few lines of code, select them with the mouse, then right-click and choose '*Evaluate Selection*' (or simply select the code and hit the shortcut *F9*).



Part 1. Gain confidence with brain connectivity matrices

In the morning lectures different processing steps, algorithms and pitfalls have been considered for the estimation of structural and functional brain connectivity networks. Furthermore, principles of graph theory and brain network characterization have been discussed. You could for example refer to the following books for a more extensive coverage of the topic: (Newman 2010), (Sporns 2016).
In this hands-on session you will investigate topological features of structural brain networks (or structural connectomes) estimated from diffusion spectrum imaging and deterministic streamline tractography. Each network is completely defined by its adjacency matrix, and it represents the macroscale white matter connectivity between 82 cortical and subcortical regions as defined by the FreeSurfer Desikan-Killiany atlas (Desikan et al., 2006).

- Load data into MATLAB:

```
>> load('Data\Connectome_HandsOn_Dataset.mat');
```

  Inspect the variables that you have just loaded by having a look at your MATLAB 'Workspace' window. Double-click on a variable to visualize it. Use the '*size*' command to inspect the dimensions of your variables, e.g.:

```
>> size(SC_ctrl)
```

What are the dimensions of the variables 'SC_ctrl' and 'SC_schz'?
What do these variables represent?

- Note that '*SC_ctrl*' contains data from healthy subjects, while '*SC_schz*' contains data from chronic schizophrenia patients. Let's now extract a single-subject connectivity matrix:

```
>> A = SC_ctrl(:,:,1);
```

  Use the command '*imagesc*' to visualize the matrix '*A*':

```
>> figure, imagesc(A), axis equal tight, colorbar;
```

(remember to check out the MATLAB documentation if you are curious to know more about MATLAB commands).

In order to gain a better visual contrast between large and small connection weights, display the connectivity matrix using a logarithmic scale:

```
>> figure, imagesc(log(A)), axis equal tight, colorbar;
>> title('log(connectivity matrix)');
```

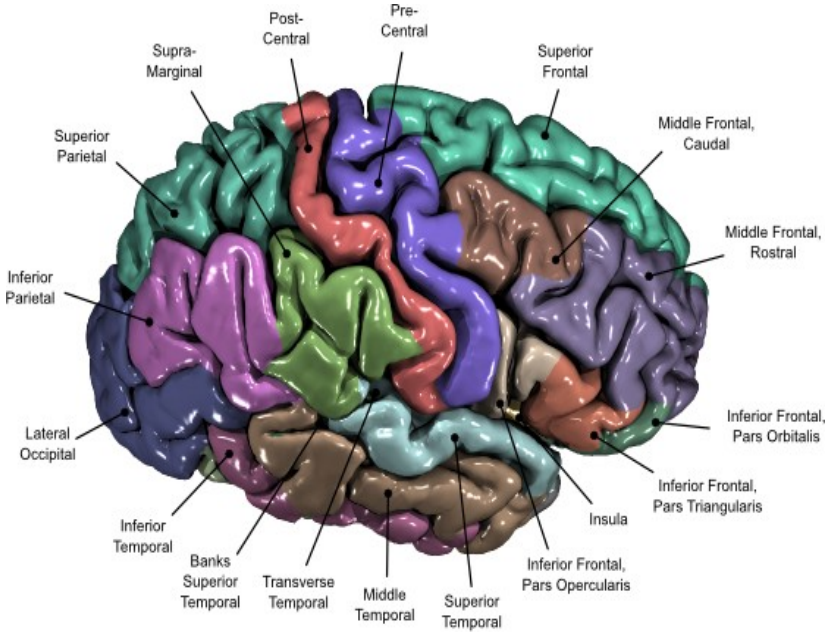
Use the function '*hist*' to visualize the histogram of the non-null connection weights:

```
>> figure, hist(A(A>0),20), xlabel('Connectivity weights');
```

Note that MATLAB allows you to filter the elements of an array (or of a matrix) by applying one or more conditions to the array. In this case, the condition '*(A>0)*' selects the elements of '*A*' which are larger than zero.

What does a row of matrix 'A' represent? Does the matrix represent a directed or indirected network? Can you identify any pattern in the matrix such as symmetries, quadrants and diagonals? (to understand the 'anatomical' organization of the matrix, have a look at the variable 'labels') Where do you find the connections with the higher and lower connection weights? Are there self-loops?

How are the connection weights distributed? Are they normally distributed? Given the range of values you observe, what could these connection weights represent?



Desikan-Killiany atlas - *https://brainder.org/tag/freesurfer/*

- The adjacency matrix you have been working on represents a weighted network. In this exercise you will convert it to an unweighted network:

```
>> th = 0;
>> B = double(A > th);
```

Note that in MATLAB the result of the operation '*(A>th)*' is a logical matrix. It is convenient to convert it to a numerical matrix using the command '*double*'.
Visualize the matrix '*B*' using '*imagesc*' and compute its network density:

```
>> nn = size(B,1); % this is the number of nodes
>> d = nnz(B) / (nn * (nn-1));
```

Note that in MATLAB the symbol '*%*' indicates a comment. '*nnz*' returns the number of non-zero elements in a matrix.

What is the meaning of the variable 'th'? Could we choose a different threshold value? Is there any relationship between the threshold value and the number of false positive and false negative connections we might expect in the network 'B'?
Is there any disconnected node in this network, for the chosen threshold value?
What is the network density? Can you explain the formula we used to compute it?

- In many situations it can be interesting to obtain a network representative of a whole group of subjects. A possible way to build a group-representative network is to select the connections that are present in at least a certain percentage of subjects over the group. To build the group-representative network of our healthy subjects we will first threshold each individual connectivity matrix:

```
>> ns = size(SC_ctrl,3); % this is the number of subjects
>> for i = 1:ns
>>     SC_ctrl_bin(:,:,i) = double(SC_ctrl(:,:,i) > th);
>> end
>> C = sum(SC_ctrl_bin,3) / ns;
```

Note that the *for-loop* allows code to be executed repeatedly: in our case the thresholding instruction is executed for each subject *i*, with *i* going from 1 to 15 (*for i = 1:ns*).

Check the size of the variables '*SC_ctrl_bin*' and '*C*'. Visualize the matrix '*C*' using the command '*imagesc*' and plot its histogram:

```
>> figure, hist(C(:),15), xlabel('Connection recurrence');
```

Note that this instruction uses the symbol '*:*' to select all the elements of matrix '*C*'.

Now build your group-representative matrix '*SC*':

```
>> th_perc = 0.5;
>> SC = double(C >= th_perc);
```

What is the meaning of each entry of the matrix 'C'? Can you comment on its structure? How many region pairs are consistently connected (or disconnected) over the 15 healthy subjects? Are there many connections that are not reproducible over subjects? Could one use the recurrence matrix 'C' for data quality control?

What does the variable 'th_perc' represent? What is the density of the network 'SC'? Do you expect it to be larger or smaller than the single-subject density values?

2. Network measures of integration and segregation: Is the brain small-world?

In this part of the lab you will compute network measures to characterize integration and segregation properties of brain structural networks. Broadly speaking, integration measures quantify “how easy it is” to go from one node of the network to another distant node of the network. Brain integration properties are thought to underline high-order brain functions such as cognitive processing and have been shown to be affected in many different disorders. Segregation measures describe the level of connectedness between ‘neighbouring’ nodes in a network, at multiple hierarchical levels. The presence of groups of highly interconnected brain regions mirrors functional specialization properties. If you want your can have a look at some review on this topic, such as (Bullmore and Sporns 2012), (Sporns, 2013).
As it has been discussed in the morning lectures, a network that combines good integration and segregation properties is called ‘*small-world*’.
So, is the brain a small-world network?

- Let’s consider the group-representative network that you have just created:

```
>> figure, imagesc(SC), axis equal tight;
>> title('Group-representative connectivity matrix');
```

  Use the Brain Connectivity Toolbox functions to compute the distance matrix associated with '*SC*':

```
>> D = distance_bin(SC);
>> figure, imagesc(D), axis equal tight, colorbar;
>> title('Distance matrix');
```

What does each entry of matrix ‘D’ represent? Do you remember the concept of shortest path?
Which are the minimum and maximum distances between two nodes in this brain network? What is the unit of measurement of the distance values? Would that be the same for a weighted network?

Do you remember the concept of network efficiency? How does it relate to the distance matrix?

- Compute the global efficiency and the average clustering coefficient of the brain network using the *BCT*. Note that the clustering coefficient is a nodal measure, so you will get a value for each one of the 82 nodes in the network. Average the nodal values to get a measure representative of the whole-network local connectedness:

```
>> Eff = efficiency_bin(SC);
>> Cl = clustering_coef_bu(SC);
>> Cl_avg = mean(Cl);
```

Can you compute yourself the network efficiency from matrix 'D'? (compare your result with the output of the BCT function 'efficiency_wei') How does the network efficiency relate to the characteristic path length of the network?
What does the clustering coefficient represent? Compute by hand the clustering coefficient of a brain region and double-check your result with the output of the BCT function 'clustering_coef_bu'. Consider for example the left transverse temporal cortex, which has numerical ID 74.
Display the 'Eff' and 'Cl_avg' values. Does the 'SC' network demonstrate high efficiency? Does it demonstrate high clustering properties? Can you interpret the efficiency and clustering values in absolute terms? Generally speaking, which basic network properties can influence the efficiency and clustering values?

- It is difficult to assess whether a network has high or low integration or segregation properties without comparing it to a reference model. In their seminal paper, Watts and Strogatz compared a series of networks to a reference random network where all connections have the same probability to be drawn (Watts and Strogatz, 1998). Here, we need to think about a good reference model to probe the brain network integration and segregation properties.

  By definition, a reference model preserves some properties of the object under investigation, while destroying some other properties of the object. Which properties of the brain network would you preserve in a fair reference model?

  Use the *BCT* to compute a reference network from the '*SC*' network and visualize it:

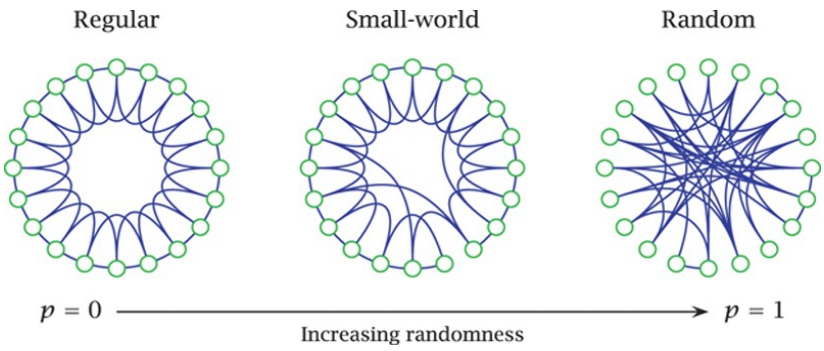
```
>> R = randmio_und(SC, 100);
>> figure, subplot(1,2,1), imagesc(SC), axis equal tight;
>> title('Brain network');
>> subplot(1,2,2), imagesc(R), axis equal tight;
>> title('Randomized network');
```

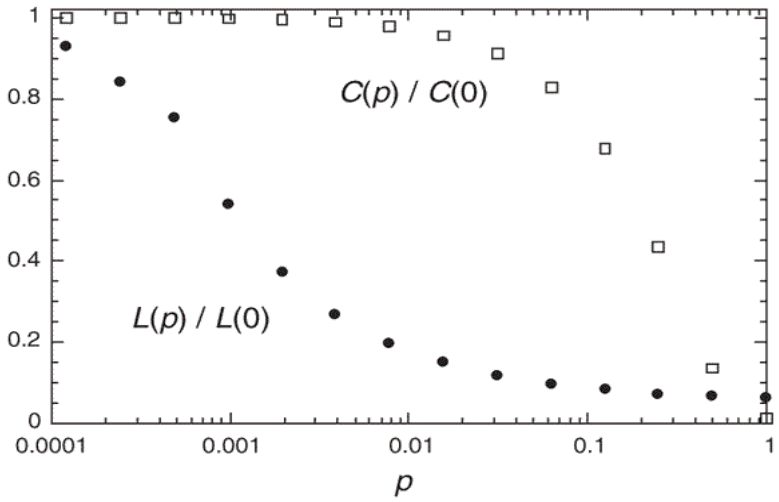
Visually compare the brain network and its randomized version. What do you observe? Can you recognize characteristic patterns (symmetries, quadrants, etc.) in the randomized network?
What is the density of the randomized network? Have the two networks the same number of nodes and the same number of connections? Pick up a network node and check its number of connections in both the brain and the randomized network. Are they different?

- Compute the characteristic path length and the clustering coefficient of the randomized network and compare them with the brain network values (for example by computing their ratio):

```
>> Cpl = sum(sum(D) / (nn-1)) / nn;
>> D_rand = distance_bin(R);
>> Cpl_rand = sum(sum(D_rand) / (nn-1)) / nn;
>> Cl_rand = clustering_coef_bu(R);
>> Cl_rand_avg = mean(Cl_rand);
>> Cpl_ratio = Cpl / Cpl_rand;
>> Cl_ratio =  Cl_avg / Cl_rand_avg;
```

Do the 'Cpl_ratio' and 'Cl_ratio' values tell us something about the brain network organization? How is the small-world index defined? Can you compute the small-world index for the brain network? Is the brain a small-world network?
Estimate a new randomized version of the brain network (using the 'randmio_und' function) and re-compute the small-world index. Do you obtain the same numerical result as before? How can you handle this issue?





(Watts and Strogatz, 1998)

Part 3. Brain hubs and network visualization

In this section we will identify the most topologically central regions in the brain network, usually referred to as network '*hubs*'. On this topic see for example (van den Heuvel and Sporns, 2011).

- Compute the nodal degrees of the group-representative network '*SC*' and visualize the degree distribution and the degree sequence of the brain network. The degree distribution is the probability distribution of the network degrees and is very important when studying network properties. The degree sequence is the monotonic non-increasing sequence of the sorted nodal degrees:

```
>> k = sum(SC); % nodal degrees
>> figure, hist(k,10), title('Degree distribution');
>> [k_sorted, k_index] = sort(k, 'descend');
>> figure, bar(k_sorted), title('Degree sequence');
>> % Set figure properties
>> xticklabel_rotate(1:nn, 90, labels(k_index));
>> set(gcf,'Units','Normalized','OuterPosition',[0 0 1 1]);
```

  Have a look at the documentation of the functions '*sum*' and '*sort*' to understand their output. Check out the size and content of the variables '*k*', '*k_sorted*' and '*k_index*'. Note that the MATLAB command '*set*' can be used to set the properties of a figure ('*gcf*' stands for 'get current figure'). Here we are setting the labels of the x-axis and the size of the figure.

Why do we use the function 'sum' to compute the degrees? How is the nodal degree defined?
Visually inspect the degree sequence of the brain network 'SC'. Have all the nodes similar degree? What is the minimum degree of a node? What is the maximum degree? Are there some nodes with degree equal to zero? What would that mean?

Re-consider the randomized network 'R'. Do you think that the degree distributions of 'R' and 'SC' will be different? Compute and visualize the degree distribution of the randomized network 'R'. What do you expect to observe?
How could you identify the hubs of the brain network? Can you list some of the brain hubs? Can you guess why a damage to a hub region (e.g., a localized stroke insult) can be particularly harmful?
Do the hub regions form a rich-club in the brain network 'SC'? Is this always the case?

- The nodal degree is only one of the possible measures of nodal centrality. Other centrality measures are for example the closeness centrality and the betweenness centrality. Indeed, you can easily compute the nodal closeness centrality using the distance matrix '*D*' that you have computed in Part 2 of the lab. The betweenness centrality is defined as the number of shortest paths in the network that pass by a given node. Different measures of centrality are inter-related but express slightly different aspects of the node topological roles in the network. You could compute different centrality measures for brain regions and assess their relationship with a correlation analysis.
  You might also be curious about weighted network measures and their relationship with binary network measures. Compute a group-representative weighted network '*SCw*' for the healthy subject group and its nodal strengths. Compare the nodal degrees and the nodal strengths:

```
>> SCw = zeros(nn);
>> for i = 1:nn
>>    for j = 1:nn
>>        this_connection = SC_ctrl(i,j,:);
>>        SCw(i,j)=mean(this_connection(this_connection>0));
>>   end
>> end
>> SCw(isnan(SCw)) = 0;
>> SCw = SCw .* SC;
>> figure, imagesc(SCw), colorbar, axis equal tight;
```

```
>> title('Weighted connectivity matrix');
>>
>> kw = sum(SCw);  % nodal strength
>> figure, plot(k,kw,'o'), grid on;
>> xlabel('Degree'), ylabel('Strength');
```

Note that the MATLAB command '*zeros*' generates (initializes) an empty matrix of size *nn* rows X *nn* columns. The command '*SC_ctrl(i,j,:)*' selects the values of the i^{th} row, j^{th} column and all the 3^{rd}-dimension elements of the 3D array '*SC_ctrl*'. The command '*.**' performs the element-wise multiplication of two matrices. In the previous code you have used two nested *for* loops: the first one walks through the rows of the matrix '*SCw*', the second one walks through its columns.

Why did you multiply the 'SCw' connectivity matrix by the binary matrix 'SC'? Why did you perform the mean on the non-zero elements of the current connection values in the for loop ('Scw(i,j) = mean(this_connection(this_connection > 0))')? Could have you chosen a different strategy?
Which kind of relationship do you expect to observe between the nodal degree and the nodal strength in a brain network? Do you think this relationship applies to all kind of networks, or does it tell you something about the structure of the brain network?

- It is often useful to visualize the results of your analyses with a 3-dimensional plots where nodes are represented as balls and connections as sticks. Visualize the brain network '*SCw*' using the BrainNet Viewer (*BNV*) toolbox. You can familiarize with the toolbox having a quick look at the online documentation or at the pdf manual that you find in the '*Lab_Brain_Connectivity_And_Graphs/Utilities/BrainNetViewer_20170403*' folder.

  Before starting the toolbox save the '*SCw*' connectivity matrix as a text file using the '*dlmwrite*' MATLAB function. Assign to the text file the extension '.edge', which is the extension recognized by the *BNV* toolbox:

```
>> dlmwrite('Data\BNV_Data\my_SCw.edge',SCw,'delimiter','\t');
```

You also need to create a second text file with extension '*.node*' containing the information about the nodes of the network. A *BNV* node file must contain 6 columns with the following data: x y z node coordinates (columns 1-3), node color (scalar values, column 4), node size (scalar values, column 5) and node labels (column 6, optional). Load into MATLAB the nodes' coordinates associated with the Desikan-Killiany parcellation and use the provided function '*generate_node_file*' to create a '*.node*' text file readable by BrainNet Viewer. You can choose which kind of information you want to encode in the node color and size attributes. For example you could color-code the network hubs, and scale the node size according to the degree:

```
>> % Color-code the node according to their 'hubness'
>> hubs = k_index(1:8);
>> n_color = zeros(nn,1);
>> n_color(hubs) = 1;
>> % Scale the size of the node according to the degree
>> n_size = k;
>> % Load node coordinates
>> load('Data\BNV_Data\centroids_bert.mat');
>> % Create .node text file
>> filepath = 'Data\BNV_Data\my_nodes.node';
>> generate_node_file(filepath, coord, n_color, n_size);
```

Now that you have generated the necessary input files, start the *BNV* toolbox. To start the BNV toolbox type 'BrainNet' at the MATLAB prompt:

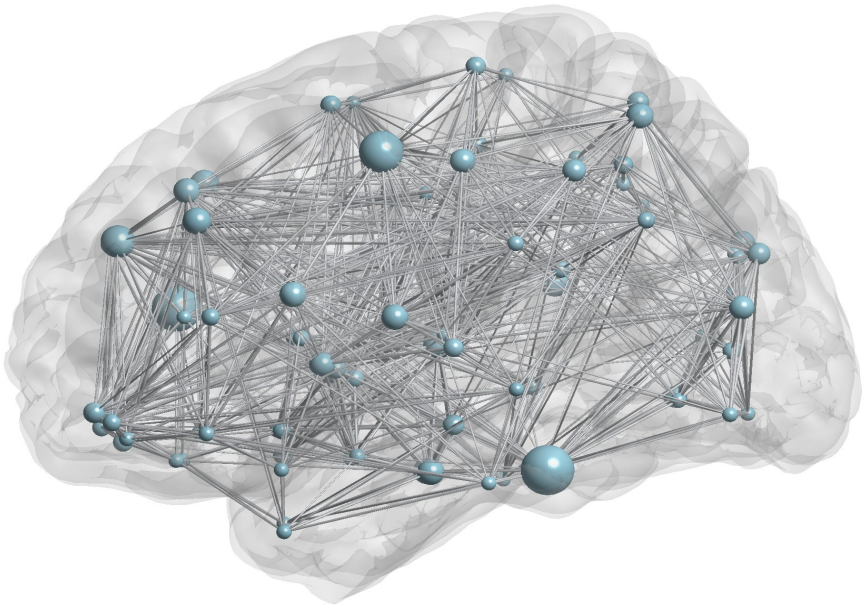
```
>> BrainNet
```

Load the '*my_nodes.node*' and '*my_SCw.edge*' files using the [*File > Load Files > Data file Browse...*] commands then press 'Ok'. Spend some time playing with the toolbox and discover its functionalities. For example you could:

- visualize the network adjacency matrix using the 'ViewMatrix' button
- change the visualization perspective using the 'Sagittal/Axial/Coronal' buttons

Do you notice any symmetry in the brain network? Can you identify the network hubs from their size? Where are the brain hubs localized (frontal / occipital / ... areas)?

- explore the node visualization options using the [*Option* > *Option* > *Node*] menu and color the network nodes according to their degree (4th column of the *.node* file) [*Node* > *Color* > *Colormap*]
- explore the edge visualization options ([*Option* > *Option* > *Edge*]). You can scale and color the network edges according to their weights



An example of brain network visualization using the BrainNet Viewer

4. Group-comparison

In this section you will apply the concepts acquired in the previous exercises to investigate brain network alterations in schizophrenia disorder. You will analyze the brain networks of schizophrenia patients (SCHZ) and healthy controls (CTRL) using the weighted global efficiency, the weighted clustering coefficient and the weighted closeness centrality measures.

The variables '*SC_schz*' and '*SC_ctrl*' contain the brain connectivity matrices of 15 chronic schizophrenia patients and 15 age- and gender-matched healthy subjects. The connectivity matrices are weighted by the connection density (not to be confounded with the network density!), defined as the number of streamlines connecting two brain regions normalized by the average streamlines length and the average surface of the connected regions (Hagmann et al., 2008). These data are part of a published dataset (Griffa et al., 2015). If you are interested to this specific topic, you might want to check out some reviews on connectomics in schizophrenia and brain disorders, for example (Fornito et al., 2017).

- Compute the weighted global efficiency of the individual SCHZ and CTRL brain networks:

```
>> % Loop over subjects
>> for i = 1:ns
>>     % CTRL
>>     this_network = SC_ctrl(:,:,i);
>>     this_network = this_network./(sum(this_network(:))/2);
>>     Eff_ctrl(i) = efficiency_wei(this_network);
>>     % SCHZ
>>     this_network = SC_schz(:,:,i);
>>     this_network = this_network./(sum(this_network(:))/2);
>>     Eff_schz(i) = efficiency_wei(this_network);
>> end
```

Read the documentation of the 'efficiency_wei' function and note the normalization of the connectivity matrix weights ('this_network = this_network./ (sum(this_network(:))/2);'). Why did we normalize the individual connectivity matrix? Could have we chosen a different strategy?

- Compare the network efficiency of the patients' and controls' groups, and check out the MATLAB documentation of the '*ranksum*' and '*ttest2*' functions.

```
>> p_Eff = ranksum(Eff_ctrl,Eff_schz);
>> [h,p_Eff_ttest] = ttest2(Eff_ctrl,Eff_schz,'tail','both');
>> figure, boxplot([Eff_ctrl',Eff_schz'],{'CTRL','SCHZ'}),
>> title('Weighted network efficiency');
```

Is the efficiency of the schizophrenia brain networks increased or decreased compared to the control group? How could you interpret this result?
What is the output of the 'ranksum' function? What is the difference between the Student's t-test and the Wilcoxon rank-sum test? Which statistical test is more appropriate in this case?

- In the previous exercise you found that the efficiency of the schizophrenia brain networks is decreased when compared to an age- and gender-matched control group. We would like now to identify which brain regions underlie this global efficiency decrease. Compute the nodal closeness centralities of the SCHZ and CTRL brain networks. Remember that the nodal closeness centrality is defined as the inverse of the average shortest path length from one node to all the other nodes in the network:

```
>> for i = 1:ns
>>     % CTRL
>>     this_network = SC_ctrl(:,:,i);
>>     this_network = this_network./(sum(this_network(:))/2);
>>     connection_lengths = 1 ./ this_network;
```

```
>>     connection_lengths(isinf(connection_lengths))= 0;
>>     [this_d,~] = distance_wei(connection_lengths);
>>     ClCent_ctrl(i,:) = 1 / (sum(this_d) / (nn-1));
>>
>>     % SCHZ
>>     this_network = SC_schz(:,:,i);
>>     this_network = this_network./(sum(this_network(:))/2);
>>     connection_lengths = 1 ./ this_network;
>>     connection_lengths(isinf(connection_lengths))= 0;
>>     [this_d,~] = distance_wei(connection_lengths);
>>     ClCent_schz(i,:) = 1 / (sum(this_d) / (nn-1));
>> end
>>
>> figure,subplot(1,2,1),hist(ClCent_ctrl(:)),title('CTRL');
>> subplot(1,2,2), hist(ClCent_schz(:)), title('SCHZ');
```

Read the documentation of the 'distance_wei' BCT function. What is the difference between a connection-weight and a connection-length matrix?
Check out the size of the variables 'ClCent_ctrl' and 'ClCent_schz'. Which values do these variable contain? How would you identify which brain regions have impaired network centrality in patients compared to control subjects? Is a multiple comparison correction needed in this case? Which multiple comparison correction methods do you know?

- Compare the closeness centralities of the 82 brain regions between the *CTRL* and *SCHZ* groups. Note that for each comparison you will get a single p-value, for a total of 82 p-values:

```
>> for i = 1:nn
>>    p_ClCent(i)=ranksum(ClCent_ctrl(:,i),ClCent_schz(:,i));
>> end
```

```
>> figure, plot(p_ClCent,'o'), hold on;
>> plot([1 nn],[0.05 0.05],'--');
>> xlabel('node'), ylabel('p-values');
>> title('Nodal p-values CTRLvsSCHZ');
```

Apply a multiple-comparison correction to limit the false discovery rate (FDR) at 0.05 probability using the function '*FDR_main*' included in the '*Utilities*' folder:

```
>> ii_survive = FDR_main(p_ClCent, 0.05, 'bh95');
>> labels(ii_survive)
```

How many brain regions have impaired closeness centrality in schizophrenia patients compared to healthy controls? Have these regions decreased or increased closeness centrality values? How many nodes present a p-value smaller than 0.05? How many nodes survive the FDR-correction?
Would you be able to apply a Bonferroni correction to the closeness centrality p-values? How many p-values survive the Bonferroni correction? What is the difference between FDR and Bonferroni correction? Which correction is more appropriate in this case?
Which are the brain regions with impaired connectivity in schizophrenia patients? Are the connectivity impairments concentrated in one lobe or hemisphere, or are they relatively spread over the whole brain network? Do you expect to observe the same alterations in a smaller/larger sample?

References

Bullmore E and Sporns O (2012) The economy of brain network organization. *Nat Rev Neurosci*, 13(5):336-349.

Desikan RS, et al. (2006) An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *Neuroimage* 31:968-980.

Fornito A, Bullmore, ET and Zalesky A (2017) Opportunities and challenges for psychiatry in the connectomic era. *Biol Psy: Cognitive Neuroscience and Neuroimaging*, 2(1):9-19.

Griffa A, et al. (2015) Characterizing the connectome in schizophrenia with diffusion spectrum imaging. *HBM* 36(1):354-366.

Hagmann P, et al. (2008) Mapping the structural core of the human cerebral cortex. *PLoS Biol* 6(7):e159.

Newman A (2010) Networks: An introduction. *Oxford University Press*, ISBN 9780199206650.

Rubinov, M. and Sporns, O. (2010). Complex network measures of brain connectivity: Uses and interpretations. *Neuroimage* 52:1059-1069.

Sporns O (2013) Network attributes for segregation and integration in the human brain. *Curr Opin Neurobiol*, 23(2):162-171.

Sporns O (2016) Networks of the brain. *The MIT Press*, ISBN: 9780262290272.

van den Heuvel M and Sporns O, 2011. Rich-club organization of the human connectome. *J Neurosci* 31(44):15775-15786.

Watts, D.J. and Strogatz, 1998. Collective dynamics of 'small-world' networks. *Nature* 393:440-442.